

EXPERIMENT FIVE


✓ Experiment 5: Familiarizing with Prolog

Problem Statement

The objective of this lab is to get hands-on experience with **Prolog**, a logic programming language, by implementing basic relationship rules and logical predicates. The tasks aim to strengthen understanding of symbolic reasoning and knowledge representation.

✓ Tasks to Complete:

- Define and query relationships using facts in Prolog.
- Create logical rules to express derived relationships.
- Implement simple predicates for common operations.
- Write predicates to convert temperature:
 - From **Celsius to Fahrenheit**
 - From **Fahrenheit to Celsius**
- Write predicates to find the:
 - **Maximum** of two numbers
 - **Minimum** of two numbers
- Translate and implement logic statements:
 1. "Everybody who has a child is happy."
 2. "If someone has a child who has a sister, then they have two children."
- Implement logic and rules to represent:
 - **Grandparent** relationships
 - **Sister** relationships

 This experiment introduces fundamental concepts of declarative programming using **facts**, **rules**, and **queries**, which are essential to symbolic AI and logical inference.

#Note: Written in Prolog (use a compatible environment to run the code)

```
% --- Family Facts ---
parent(pam, bob).
parent(tom, bob).
parent(tom, liz).
parent(bob, ann).
parent(bob, pat).
parent(pat, jim).

female(pam).
female(liz).
female(ann).
female(pat).

male(tom).
male(bob).
male(jim).

% --- Relationships ---
mother(X, Y) :- parent(X, Y), female(X).
father(X, Y) :- parent(X, Y), male(X).
child(X, Y) :- parent(Y, X).
sibling(X, Y) :- parent(Z, X), parent(Z, Y), X \= Y.
sister(X, Y) :- sibling(X, Y), female(X).
grandparent(X, Y) :- parent(X, Z), parent(Z, Y).

% --- Custom Rules ---
happy(X) :- parent(X, _).
hastwochildren(X) :- parent(X, C), sister(S, C), parent(X, S), C \= S.

% --- Simple Predicates ---
max(X, Y, X) :- X >= Y.
max(X, Y, Y) :- Y > X.

min(X, Y, X) :- X <= Y.
min(X, Y, Y) :- Y < X.
```

```
% --- Temperature Conversion ---  
c_to_f(C, F) :- F is (C * 9/5) + 32.  
f_to_c(F, C) :- C is (F - 32) * 5/9.
```